

# Distributed Event Localization via Alternating Direction Method of Multipliers

Chunlei Zhang and Yongqiang Wang

**Abstract**—This paper addresses the problem of distributed event localization using noisy range measurements with respect to sensors with known positions. Event localization is fundamental in many wireless sensor network applications such as homeland security, law enforcement, and environmental studies. However, a distributed algorithm is still lacking which can split the intensive localization computation among a wireless sensor network in which individual sensors only have limited computational capacity. Based on the alternating direction method of multipliers (ADMM), we propose a distributed event localization structure and two scalable distributed algorithms named GS-ADMM and J-ADMM respectively. More specifically, we consider a scenario in which the entire sensor network is divided into several clusters with a cluster head collecting measurements within each cluster and performing local localization. In the meantime, the cluster heads exchange intermediate computation information which will be factored into their local computations to achieve consistency (consensus) across the localization results of all cluster heads. Simulation results confirm that the proposed approaches can indeed achieve localization consistency (consensus) across the clusters and each cluster can obtain better localization performance compared with the case in which cluster heads only use local measurements within clusters.

## I. INTRODUCTION

With the ability to transmit/receive information and fuse data, smart sensors enabled and greatly advanced numerous applications such as environmental monitoring [1], target tracking [2], underwater detection [3], and acoustic gunfire localization [4], [5]. Among these applications, event localization is a significant and essential component or even the ultimate goal. Taking the military field as an example, if some threat sources or impulsive events (e.g., shooting or explosion) are detected, it is of great importance to locate these threat sources to make prompt reactions (e.g., giving warning, providing aid). A typical example is the PinPoint<sup>TM</sup> mobile sensors from BioMimetic System Inc., which can be deployed as a mobile infrastructure for impulsive threat event detection and localization [6]. PinPoint<sup>TM</sup> sensors localize an acoustic event using a small microphone array which is cost-effective, omnidirectional, and small in size. They range impulsive acoustic events by measuring the difference in time of arrival (TOA) among the microphones. In fact, since a sensor itself has integrated a microphone array, it is able to identify and locate a target without assistance or cooperation with others. However, due to close distance between the microphone cells, its accuracy is very limited and

unsatisfactory. Collaboration among the sensors is necessary to improve localization accuracy.

The above observation motivated us to provide a localization structure in which an entire network is divided into several clusters. A cluster head (which can be a regular sensor) collects and fuses information from all cluster members. Two cluster heads in different clusters can exchange information if a communication link is available between them; otherwise they don't have access to each other's information. We assume that the cluster heads form a connected network, i.e., there is a (multi-hop) path (constructed using a series of communication links connected in succession) between every pair of cluster heads. This structure provides several key desired properties for event localization: 1) Robustness to inter-cluster communication failures — if a cluster is isolated from the rest of the network due to communication failures between the cluster heads, then each cluster can still conduct event localization locally; 2) Robustness to sensor failures — if several sensors in a cluster fail, they will not completely deprive the cluster head from performing localization since the cluster head can also get information from neighboring clusters. However, this situation will be problematic in cases where each cluster conducts localization independently because normally a cluster only has a limited number of sensors; 3) Improved performance with a moderate communication overhead — by only exchanging limited intermediate optimization results rather than raw measurement data, performance can be improved compared with localization with intra-cluster measurements. This hierarchical and parallel structure has been employed in some other fields. A typical example is wide-area monitoring and control in large-scale power systems [7]. To estimate the electro-mechanical oscillation modes, a large number of phasor measurement units (PMU) have to be deployed across a network to conduct measurements. The measurements from all the PMUs have to be fused to diagnosis the inter-area oscillation modes. However, wide-area communication between PMUs is very expensive [8]. To fuse information across all the PMUs without imposing heavy communication overhead, a similar structure as ours is adopted in [7].

Sensor network based event localization is not a new problem and plenty of results are available in the literature, using either angle-of-arrival measurements [9], [10], [11], [12], or time-of-arrival (including time-difference-of-arrival) measurements [13], [14], [15]. A straightforward approach is to gather (noisy) measurements (e.g., range measurements, sensor positions) obtained by all sensors to a processing center, which then estimates the event location using a certain centralized optimization algorithm. Two classical such algorithms are semi-definite programming (SDP) [16] and multidimensional scaling (MDS-MAP) [17]. The SDP based localization

\*The work was supported in part by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001.

Chunlei Zhang and Yongqiang Wang are with the department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA {chunleiz, yongqiw}@clemson.edu

algorithm was first proposed by Doherty and coauthors in [16]. In this algorithm, linear matrix inequalities (LMIs) are adopted to present the geometric constraints between sensors, which are then combined into a single semidefinite program. This centralized algorithm will be adopted in our proposed algorithms when solving the subproblems of ADMM in each cluster, which will be detailed later. The other centralized algorithm MDS-MAP was proposed by Shang and coauthors in [17]. By adopting an information visualization technique called multidimensional scaling (MDS), MDS-MAP reconstructs the positions of sensor nodes based on given noisy distance between pairwise nodes of a network.

However, a severe shortcoming of centralized localization algorithms is that the computation complexity at the processing center might be quite high which poses great challenges for low-cost sensor nodes with limited computational capabilities. In addition, the required communications to collect all measurements to a single central node may be problematic due to possible traffic bottleneck and severe constraints on communication ranges. Moreover, once the central node fails due to, e.g., attacks or power depletion, the whole network will slip into a state of paralysis. Therefore, techniques solving the event localization problem in a distributed way are crucial for wireless sensor network based event localization.

In contrast to centralized algorithms, distributed localization algorithms are designed to run the computation over the whole network instead of on a center. In general, distributed algorithms are often established on massive parallelism or sequential calculations and mutual collaboration [18]. So compared with centralized algorithms, distributed designs have better scalability, flexibility, and robustness. Several distributed concepts have been proposed such as beacon-based distributed algorithms [19], relaxation-based distributed algorithms [20], and coordinate system stitching [21], [22], [23]. These localization algorithms are mainly used for triangulating the positions of sensors (whose positions are unknown) based on noisy distance measurements with respect to sensors with known positions (normally called anchor sensor nodes). In this paper, we consider a different problem where the positions of sensors are known and the position of a target event is unknown. Furthermore, the target event has no communication or computation capability. So specifically for our event localization structure, the key idea of distributed design is to separate the whole network into a number of clusters with each cluster composed of several sensors with known positions (also called anchor sensors in the literature). Measurements within a cluster are processed simultaneously. The computation procedure at each cluster is affected by the computational results of neighboring clusters so that all clusters can reach consistency (consensus) on the target event location finally.

The core of our distributed localization algorithms is the alternating direction method of multipliers (ADMM), which has been proven extremely suitable in distributed convex optimization, especially for large-scale problems [24]. The key idea of ADMM is to obtain a large global solution through cooperation of small local subproblems. ADMM is easy to parallelize and implement and robust to noise and computation error [25]. Our proposed cluster based localization structure

takes full advantages of ADMM which enables local optimizations within individual clusters as subproblems. Then through cooperation of subproblems in neighboring clusters, a global estimated location could be reached. That is to say, the estimated locations obtained by individual clusters are made as consistent as possible. Such consistency is of crucial importance in many applications and domains. For example, when a system is monitored and controlled by several parties, the decision obtained by each party should be in accordance with each other to avoid system failure. Additionally, when a sporadic impulsive event requiring immediate responsive actions is detected by several monitors, consistency in the estimated location from each monitor is the key for the monitors to coordinate cooperative operations.

*Contribution:* The first contribution of this paper is a distributed event localization structure which has potential for wide applications. Based on this event localization structure, two distributed localization algorithms, i.e., GS-ADMM and J-ADMM, are proposed based on ADMM. The two localization algorithms can achieve better performance than individual cluster based localization and, at the same time, guarantee consistency of the estimation across clusters with limited communications between clusters. In addition, the algorithms are proven to converge with a convergence rate of  $O(1/t)$  ( $t$  is the iteration time).

*Organization:* The rest of this paper is organized as follows: Section II states the formulation of the problem. To solve the problem, a convex relaxation is required and the method proposed by [25] is recapitulated in Section III. In Section IV, two algorithms named GS-ADMM and J-ADMM based on ADMM are proposed, with their convergence properties analyzed in Section V. Section VI gives numerical simulation results. In the end, a conclusion is made in Section VII.

## II. PROBLEM STATEMENT

Consider a localization sensor network divided into  $m$  clusters (cf. Fig. 1 for the case  $m = 4$ ). Denote the number of constituent sensors of cluster  $i$  as  $N_i$  ( $i = 1, 2, \dots, m$ ). We consider localization in  $D$  ( $D \in \{1, 2, 3\}$ ) dimensional Euclidean space and suppose the location of event is denoted as  $\mathbf{x} \in \mathbb{R}^D$ . Denote the position of the  $k$ th sensor in the  $i$ th cluster as  $\mathbf{a}_{i,k} \in \mathbb{R}^D$ . And the  $k$ th sensor in the  $i$ th cluster can obtain a noisy range measurement  $r_{i,k}$  of its distance with respect to a target event:

$$r_{i,k} = d_{i,k} + v_{i,k}$$

where  $d_{i,k} = \|\mathbf{x} - \mathbf{a}_{i,k}\|$  denotes the actual distance between the event position and the  $k$ th sensor of the  $i$ th cluster and  $v_{i,k}$  is the Gaussian noise term.

Then the event localization problem amounts to estimating the unknown event location  $\mathbf{x}$  using known sensor positions  $\mathbf{a}_{i,k}$  and noisy range measurements  $r_{i,k}$  ( $i = 1, 2, \dots, m, k = 1, 2, \dots, N_i$ ). As shown in Fig. 1, we assume that each cluster has a cluster head in each cluster  $i$ , which can gather all range measurements  $r_{i,k}$  from sensors within the cluster. In addition, a cluster head can communicate and exchange information with the cluster head of a neighboring cluster if there is a

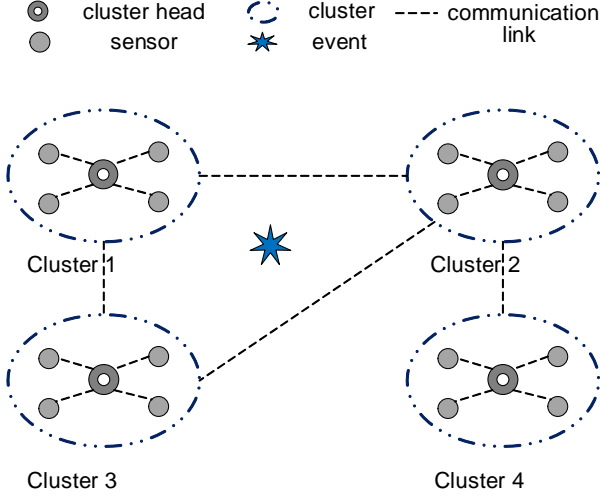


Fig. 1: Cluster based event localization structure ( $m = 4$ )

communication link between them (cf. Fig. 1). In this case, we also say that these two clusters can communicate. We assume that the communication pattern forms a connected network, i.e., there is a (multi-hop) path (composed of multiple communication links connected in succession) between any pair of cluster heads. For example, in Fig. 1, cluster 1 is able to interchange information with cluster 2 and 3 (via cluster heads); cluster 2 can exchange information with cluster 1, 3 and 4 (via cluster heads), etc. Denote  $B_i$  as the set of all neighboring clusters of cluster  $i$  and  $\hat{B}_i$  as the union of set  $B_i$  and cluster  $i$  itself.

As in most existing results, we also use the maximum likelihood method (ML) for network localization [25], [26]. Let  $p_{i,k}(d_{i,k}(\mathbf{x}, \mathbf{a}_{i,k})|r_{i,k})$  denote the measuring probability density function (PDF) for sensor  $k$  in cluster  $i$ , we can define the localization problem as follows:

**Problem Statement:** The event localization problem is to obtain the position of  $\mathbf{x} \in \mathbb{R}^D$ , with both  $r_{i,k}$  and  $\mathbf{a}_{i,k}$  given.

Under the assumption of zero-mean additive Gaussian noise, we can rewrite this problem using the maximum likelihood (ML) method:

$$\mathbf{x}_{\text{ML}}^* = \underset{\mathbf{x} \in \mathbb{R}^D}{\operatorname{argmax}} \sum_{i=1}^m \sum_{k=1}^{N_i} \ln p_{i,k}(d_{i,k}(\mathbf{x}, \mathbf{a}_{i,k})|r_{i,k}) \quad (1)$$

Problem (1) is a non-convex optimization problem. To solve non-convex optimization problem, a direct and efficient way is adopting some convex relaxation techniques to transform it into a convex approximate formulation. Typical techniques conclude semidefinite programming (SDP) relaxation [27], [28], [29], second-order cone programming relaxation (SOCP) [30], sum of squares (SOS) [31] convex relaxation, and maximum likelihood (ML) based relaxation [14], [32]. To this end, we first make one assumption, as in most localization approaches [25]:

**Assumption 1:** The measuring PDF  $p_{i,k}(d_{i,k}(\mathbf{x}, \mathbf{a}_{i,k})|r_{i,k})$  is a log-concave function of unknown distance  $d_{i,k}$ .

### III. CONVEX RELAXATION

Problem (1) is non-convex and it is generally infeasible to find a global solution [25]. In this paper, following the idea of [25], we introduce the following convex relaxation approach to convert problem (1) to a convex optimization problem.

To facilitate the conversion, we first define the following new variables:  $y = \mathbf{x}^T \mathbf{x}$ ,  $\varepsilon_{i,k} = d_{i,k}^2$ . Then we stack  $\varepsilon_{i,k}, k \in \{1, 2, \dots, N_i\}$  into  $\boldsymbol{\varepsilon}_i$  and further  $\boldsymbol{\varepsilon}_i, i \in \{1, 2, \dots, m\}$  into  $\boldsymbol{\varepsilon} \triangleq [\boldsymbol{\varepsilon}_1^T, \boldsymbol{\varepsilon}_2^T, \dots, \boldsymbol{\varepsilon}_m^T]^T$ . In the same way we stack  $d_{i,k}$  into  $\mathbf{d}_i$  and  $\mathbf{d} \triangleq [\mathbf{d}_1^T, \mathbf{d}_2^T, \dots, \mathbf{d}_m^T]^T$ . Then the cost function can be written as

$$f(\mathbf{d}) = - \sum_{i=1}^m \sum_{k=1}^{N_i} \ln p_{i,k}(d_{i,k}|r_{i,k}) \quad (2)$$

Consider the case of white zero-mean Gaussian noise, i.e.,  $v_{i,k} \sim \mathcal{N}(0, \sigma_{i,k}^2)$ , then problem (2) can be rewritten as

$$\begin{aligned} f(\mathbf{d}) &= \sum_{i=1}^m \sum_{k=1}^{N_i} \sigma_{i,k}^{-2} (d_{i,k}^2 - 2d_{i,k}r_{i,k} + r_{i,k}^2) \\ &= \sum_{i=1}^m \sum_{k=1}^{N_i} \sigma_{i,k}^{-2} (\varepsilon_{i,k} - 2d_{i,k}r_{i,k} + r_{i,k}^2) \end{aligned} \quad (3)$$

Now, problem (1) can be relaxed into the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{x}, \boldsymbol{\varepsilon}, \mathbf{d}} \quad & f(\mathbf{d}) \\ \text{subject to} \quad & y - 2\mathbf{x}^T \mathbf{a}_{i,k} + \|\mathbf{a}_{i,k}\|^2 = \varepsilon_{i,k}, \quad y = \mathbf{x}^T \mathbf{x} \\ & \varepsilon_{i,k} = d_{i,k}^2, \quad d_{i,k} \geq 0 \\ & \forall i \in \{1, 2, \dots, m\}, \quad k \in \{1, 2, \dots, N_i\} \end{aligned} \quad (4)$$

However, in this case, the constraints of (4) are still combined into a non-convex set. Using Schur complements [33], the following convex relaxation can be obtained:

$$\begin{aligned} \min_{\mathbf{x}, \boldsymbol{\varepsilon}, \mathbf{d}} \quad & f(\mathbf{d}) \\ \text{subject to} \quad & y - 2\mathbf{x}^T \mathbf{a}_{i,k} + \|\mathbf{a}_{i,k}\|^2 = \varepsilon_{i,k}, \quad \varepsilon_{i,k} \geq 0 \\ & \begin{pmatrix} 1 & d_{i,k} \\ d_{i,k} & \varepsilon_{i,k} \end{pmatrix} \succeq 0, \quad d_{i,k} \geq 0 \\ & \forall i \in \{1, 2, \dots, m\}, \quad k \in \{1, 2, \dots, N_i\} \\ & \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x}^T & y \end{pmatrix} \succeq 0, \quad y \geq 0 \end{aligned} \quad (5)$$

Problem (5) is a convex problem with inequality constraints [24], and more specifically, a semidefinite programming (SDP) problem. So in the following section, we can propose ADMM based algorithms to solve this problem.

### IV. PROPOSED DISTRIBUTED ALGORITHMS

#### A. Fundamentals: Standard ADMM

ADMM is an algorithm which is suitable to solve problems in the following form [24]:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} + \mathbf{K}\mathbf{z} = \mathbf{c} \end{aligned} \quad (6)$$

where  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{z} \in \mathbb{R}^m$ ,  $\mathbf{A} \in \mathbb{R}^{p \times n}$ ,  $\mathbf{K} \in \mathbb{R}^{p \times m}$  and  $\mathbf{c} \in \mathbb{R}^p$ , and  $f(\mathbf{x})$  and  $g(\mathbf{z})$  are convex functions. To get the optimal value

$p^* = \inf\{f(\mathbf{x}) + g(\mathbf{z}) \mid \mathbf{Ax} + \mathbf{Kz} = \mathbf{c}\}$  for problem (6), first an augmented Lagrangian function is formed as follows:

$$\mathcal{L}_\rho(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}) = f(\mathbf{x}) + g(\mathbf{z}) + \boldsymbol{\mu}^T (\mathbf{Ax} + \mathbf{Kz} - \mathbf{c}) + \frac{\rho}{2} \|\mathbf{Ax} + \mathbf{Kz} - \mathbf{c}\|^2$$

where  $\boldsymbol{\mu}$  is the Lagrange multiplier associated with the constraint  $\mathbf{Ax} + \mathbf{Kz} = \mathbf{c}$  and  $\rho > 0$  is a predefined penalty parameter. Then ADMM solves problem (6) by updating  $\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}$  in the following sequence: firstly an  $\mathbf{x}$ -minimization step (7), then a  $\mathbf{z}$ -minimization step (8), and finally a dual variable update (9):

$$\mathbf{x}^{k+1} = \operatorname{argmin}_{\mathbf{x}} \mathcal{L}_\rho(\mathbf{x}, \mathbf{z}^k, \boldsymbol{\mu}^k) \quad (7)$$

$$\mathbf{z}^{k+1} = \operatorname{argmin}_{\mathbf{z}} \mathcal{L}_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \boldsymbol{\mu}^k) \quad (8)$$

$$\boldsymbol{\mu}^{k+1} = \boldsymbol{\mu}^k + \rho(\mathbf{Ax}^{k+1} + \mathbf{Kz}^{k+1} - \mathbf{c}) \quad (9)$$

In the following part, we will propose two distributed algorithms for event localization using the framework of standard ADMM.

### B. Problem Reformulation

In multi-node distributed implementation algorithms, neighboring nodes have to generate and exchange copies of local estimated values to ensure a consistent global estimation across all local nodes. In our event localization structure, a cluster is treated as a normal node which solves a common sensor network localization problem based on measurements obtained by sensors within the cluster. And neighboring clusters exchange intermediate computational results (through cluster heads) to guarantee that all clusters reach the same estimation value. In some sense, the way we separate the entire network into clusters is close to heuristic algorithms, which also divide all nodes of a large-scale network into arbitrary clusters. But in our algorithms, clusters exchange information and are affected by neighboring clusters when exploring their local optimal estimations, whereas heuristic algorithms just patch together solutions from individual clusters [25].

To better interpret our algorithms, we define a local vector

$$\mathbf{p}_i \triangleq (\boldsymbol{\varepsilon}_i^T, \mathbf{d}_i^T, y_i, \mathbf{x}_i^T)^T \in \mathbb{R}^{2N_i+D+1}, \quad i \in \{1, 2, \dots, m\}$$

which is owned by cluster  $i$ .

We let  $\mathbf{p}$  denote the stacked vector of  $\mathbf{p}_i$  and define a convex set

$$\mathcal{P}_i \triangleq \{\mathbf{p}_i \mid \mathbf{p}_i \text{ verifies (5)}\}$$

Then problem (5) can be rewritten as

$$\begin{aligned} \min_{\mathbf{p}} \quad & f(\mathbf{p}) \\ \text{subject to} \quad & \mathbf{p}_i \in \mathcal{P}_i, \quad \forall i \in \{1, 2, \dots, m\} \end{aligned} \quad (10)$$

where, in our situation,  $f(\mathbf{p})$  is given as follows:

$$f(\mathbf{p}) = - \sum_{i=1}^m \sum_{k=1}^{N_i} \ln p_{i,k}(d_{i,k} | r_{i,k}) = \sum_{i=1}^m f_i(\mathbf{p}_i) \quad (11)$$

### C. ADMM based solutions

For the structure in (11), it is easy to see that problem (10) can be divided into  $m$  subproblems. And then it can be solved in a distributed way using ADMM by adding some constraints on  $\mathbf{p}_i$ . Next we present the basic idea based on a graph theory based formulation of the communication pattern.

Using graph theory [34], the communication pattern of the cluster heads can be represented by  $G = \{V, E\}$ , where the set  $V$  denotes the set of cluster heads, and  $E$  denotes the set of undirected edges (communication links) between clusters. We use  $e_{i,j} \in E, i < j$  to denote the link (if there is) between cluster heads  $i$  and  $j$ . We use  $|E|$  to represent the total number of undirected edges. In our problem formulation, each cluster is associated with a local cost function  $f_i(\mathbf{p}_i)$ , and all clusters work together to solve the problem in (10). Assume that the local cost function  $f_i$  is only known to cluster  $i$ , then to reach consistency (consensus) of estimated position values among all clusters, we impose constraint  $\mathbf{x}_i = \mathbf{x}_j$  if there exists an edge  $e_{i,j} \in E$  between clusters  $i$  and  $j$ . Introduce a matrix  $J_i = [0_{D \times (2N_i+1)}, I_D] \in \mathbb{R}^{D \times (2N_i+D+1)}$ , where  $I_D$  denotes the  $D$  dimensional identity matrix, then  $\mathbf{x}_i$  can be presented as  $\mathbf{x}_i = J_i \mathbf{p}_i$ , so the constraint  $\mathbf{x}_i = \mathbf{x}_j$  can be presented as  $J_i \mathbf{p}_i = J_j \mathbf{p}_j$

Now we are able to rewrite problem (10) into a distributed ADMM form as follows:

$$\begin{aligned} \min_{\mathbf{p}_i, i \in \{1, 2, \dots, m\}} \quad & \sum_{i=1}^m f_i(\mathbf{p}_i) \\ \text{subject to} \quad & J_i \mathbf{p}_i = J_j \mathbf{p}_j, \quad \forall i \in \{1, 2, \dots, m\}, j \in B_i \\ & \mathbf{p}_i, \mathbf{p}_j \in \mathcal{P}_i \end{aligned} \quad (12)$$

or in a more compact way:

$$\begin{aligned} \min_{\mathbf{p}} \quad & f(\mathbf{p}) \\ \text{subject to} \quad & C\mathbf{p} = 0, \quad \mathbf{p}_i \in \mathcal{P}_i, \quad \forall i \in \{1, 2, \dots, m\} \end{aligned} \quad (13)$$

where  $\mathbf{p} = [\mathbf{p}_1^T, \mathbf{p}_2^T, \dots, \mathbf{p}_m^T]^T$ ,  $J = \operatorname{diag}\{J_1, J_2, \dots, J_m\} \in \mathbb{R}^{mD \times (\sum_{i=1}^m 2N_i + D + 1)}$ , and  $C$  is the edge-node incidence matrix of graph  $G$  as defined in [35]. For example, in the one-dimensional case ( $D = 1$ ),  $C = [c_{i,j}]$  is an  $|E| \times m$  matrix whose  $|E|$  rows correspond to the  $|E|$  edges and the  $m$  columns correspond to the  $m$  clusters such that:

$$c_{i,j} = \begin{cases} 1 & \text{if the } i^{\text{th}} \text{ edge originates at cluster } j \\ -1 & \text{if the } i^{\text{th}} \text{ edge terminates at cluster } j \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

Here we define that each edge  $e_{i,j}$  originates at  $i$  and terminates at  $j$  (note:  $e_{i,j} \in E, i < j$ ).

It can easily be verified that the matrix  $C$  for Fig. 1 is

$$C = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad (15)$$

For high dimensional cases, where  $D \geq 2$ ,  $C \in \mathbb{R}^{|E| \times mD}$  can be obtained by replacing the value of 1 and  $-1$  with  $I_D$

and  $-I_D$  with  $I_D$  denoting the  $D$  dimensional identity matrix. Then the  $C$  matrix for Fig. 1 becomes

$$C = \begin{bmatrix} I_D & -I_D & 0_D & 0_D \\ 0_D & I_D & -I_D & 0_D \\ I_D & 0_D & -I_D & 0_D \\ 0_D & I_D & 0_D & -I_D \end{bmatrix} \quad (16)$$

In this formulation, after each cluster obtains its local estimation  $\mathbf{p}_i$ , it sends the value  $J_i \mathbf{p}_i$  (estimated event position  $\mathbf{x}_i$ ) to neighboring clusters. By adding the constraint  $J_i \mathbf{p}_i = J_j \mathbf{p}_j, \forall i \in \{1, 2, \dots, m\}, j \in B_i$  as shown in (12), the consistency of individual local estimate event position  $J_i \mathbf{p}_i(\mathbf{x}_i)$  obtained by each cluster is guaranteed. Now we are in place to present our detailed algorithms to solve (12).

#### D. Proposed Algorithms

Let  $\lambda_{i,j}$  be the Lagrange multiplier relevant to the constraint  $J_i \mathbf{p}_i = J_j \mathbf{p}_j$ . Then the regularized augmented Lagrangian function of problem (12) is denoted as

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{p}, \boldsymbol{\lambda}) = & \sum_{i=1}^m f_i(\mathbf{p}_i) \\ & + \sum_{i=1}^m \sum_{j \in B_i} (\lambda_{i,j}^T (J_i \mathbf{p}_i - J_j \mathbf{p}_j)) + \sum_{i=1}^m \sum_{j \in B_i} \frac{\rho}{2} \|J_i \mathbf{p}_i - J_j \mathbf{p}_j\|^2 \end{aligned} \quad (17)$$

where  $\lambda_{i,j}$  are stacked into  $\boldsymbol{\lambda}_i$  for all  $j \in B_i$  and  $\boldsymbol{\lambda}_i$  are stacked into  $\boldsymbol{\lambda}$  for all  $i \in \{1, 2, \dots, m\}$ . Let

$$\begin{aligned} \mathcal{L}_i(\mathbf{p}_i, \boldsymbol{\lambda}_i) = & f_i(\mathbf{p}_i) \\ & + \sum_{j \in B_i} (\lambda_{i,j}^T (J_i \mathbf{p}_i - J_j \mathbf{p}_j)) + \frac{\rho}{2} \|J_i \mathbf{p}_i - J_j \mathbf{p}_j\|^2 \end{aligned} \quad (18)$$

Then we can rewrite (17) as

$$\mathcal{L}_\rho(\mathbf{p}, \boldsymbol{\lambda}) = \sum_{i=1}^m \mathcal{L}_i(\mathbf{p}_i, \boldsymbol{\lambda}_i) \quad (19)$$

It is clear from (19) that problem (17) can be decoupled across individual clusters  $i, i \in \{1, 2, \dots, m\}$ . Therefore, each cluster  $i$  considers an individual augmented Lagrangian function  $\mathcal{L}_i(\mathbf{p}_i, \boldsymbol{\lambda}_i)$ . So for cluster  $i$ , the updating recursion is:

$$\mathbf{p}_i^{t+1} = \operatorname{argmin}_{\mathbf{p}_i \in \mathcal{P}} \mathcal{L}_i(\mathbf{p}_i, \boldsymbol{\lambda}_i^t) \quad (20)$$

$$\boldsymbol{\lambda}_{i,j}^{t+1} = \boldsymbol{\lambda}_{i,j}^t + \rho (J_i \mathbf{p}_i^{t+1} - J_j \mathbf{p}_j^{t+1}) \quad (21)$$

Here, we can update  $\mathbf{p}_i$  in two different ways. One way is based on the Gauss-Seidel update [36], in which clusters update in a sequential order. The other way is the Jacobian scheme in which all clusters update in parallel [37].

**GS-ADMM:** We first consider an algorithm based on the Gauss-Seidel update. Gauss-Seidel update for distributed ADMM has been explored theoretically and proven to be able to converge in most cases for convex objective functions (see, e.g., [38], [39], [40]). GS-ADMM based solution for distributed event localization can be described as follows:

---

#### Algorithm I: GS-ADMM

---

Set  $\mathbf{p}_i^0, \boldsymbol{\lambda}_{i,j}^0$  to zero for each cluster.

**Input:**  $\mathbf{p}_i^t, \boldsymbol{\lambda}_{i,j}^t$

**Output:**  $\mathbf{p}_i^{t+1}, \boldsymbol{\lambda}_{i,j}^{t+1}$

- 1) All clusters update their local vectors in a sequential order and send their local vectors  $J_i \mathbf{p}_i^{t+1}$  to neighboring clusters  $B_i$  immediately, where

$$\begin{aligned} \mathbf{p}_i^{t+1} = & \operatorname{argmin}_{\mathbf{p}_i \in \mathcal{P}} f_i(\mathbf{p}_i) + \\ & \sum_{j \in \hat{B}_i, j \geq i} (\lambda_{i,j}^{tT} (J_i \mathbf{p}_i - J_j \mathbf{p}_j^t) + \frac{\rho}{2} \|J_i \mathbf{p}_i - J_j \mathbf{p}_j^t\|^2) + \\ & \sum_{j \in \hat{B}_i, j < i} (\lambda_{i,j}^{tT} (J_i \mathbf{p}_i - J_j \mathbf{p}_j^{t+1}) + \frac{\rho}{2} \|J_i \mathbf{p}_i - J_j \mathbf{p}_j^{t+1}\|^2) \end{aligned} \quad (22)$$

Here we also consider the effect of  $J_i \mathbf{p}_i^t$  when updating  $\mathbf{p}_i^{t+1}$  by adding a term  $\frac{\rho}{2} \|J_i \mathbf{p}_i - J_i \mathbf{p}_i^t\|^2$ . Problem (22) is an SDP problem that can be solved by common convex toolboxes such as Yalmip [25], [41], which is used in our simulations.

- 2) Each cluster computes

$$\boldsymbol{\lambda}_{i,j}^{t+1} = \boldsymbol{\lambda}_{i,j}^t + \rho (J_i \mathbf{p}_i^{t+1} - J_j \mathbf{p}_j^{t+1}) \quad (23)$$

- 3) Set  $t = t + 1$ , back to 1).
- 

In GS-ADMM, all clusters update their local estimated position values in a sequential way, which requires a global predefined order and is not amenable for parallelization. To overcome this shortcoming, we also propose another algorithm based on Jacobian scheme.

**J-ADMM:** Algorithm J-ADMM is motivated by the work in [42], which proposed Proximal Jacobian ADMM by adding some proximal terms when updating  $\mathbf{p}_i$ . We adopt the same idea here and prove that if the proximal terms meet some additional requirements, convergence of this algorithm can be guaranteed. The analysis about convergence is detailed in the following section. The detailed procedure of J-ADMM is given as follows:

---

#### Algorithm II: J-ADMM

---

Set  $\mathbf{p}_i^0, \boldsymbol{\lambda}_{i,j}^0$  to zero for each cluster.

**Input:**  $\mathbf{p}_i^t, \boldsymbol{\lambda}_{i,j}^t$

**Output:**  $\mathbf{p}_i^{t+1}, \boldsymbol{\lambda}_{i,j}^{t+1}$

- 1) Each cluster updates its local vector in parallel:

$$\begin{aligned} \mathbf{p}_i^{t+1} = & \operatorname{argmin}_{\mathbf{p}_i \in \mathcal{P}} f_i(\mathbf{p}_i) \\ & + \sum_{j \in B_i} (\lambda_{i,j}^{tT} (J_i \mathbf{p}_i - J_j \mathbf{p}_j^t) + \frac{\rho}{2} \|J_i \mathbf{p}_i - J_j \mathbf{p}_j^t\|^2) \\ & + \frac{\rho \gamma_i}{2} \|J_i \mathbf{p}_i - J_i \mathbf{p}_i^t\|^2 \end{aligned} \quad (24)$$

The last term of the above equality, i.e.,  $\frac{\rho \gamma_i}{2} \|J_i \mathbf{p}_i - J_i \mathbf{p}_i^t\|^2$ , is the proximal term we added where  $\gamma_i \geq 0$  is a scalar. This is an SDP problem and can be solved using common convex toolboxes such as Yalmip [25], [41], which is used in our simulations.

- 2) Each cluster sends its local vector  $J_i \mathbf{p}_i^{t+1}$  to neighboring clusters in  $B_i$ .
- 3) Each cluster computes

$$\boldsymbol{\lambda}_{i,j}^{t+1} = \boldsymbol{\lambda}_{i,j}^t + \rho (J_i \mathbf{p}_i^{t+1} - J_j \mathbf{p}_j^{t+1}) \quad (25)$$

- 4) Set  $t = t + 1$ , back to 1).
-

*Remark 1:* A distinct difference between GS-ADMM and J-ADMM is the way they update  $\mathbf{p}_i$ . In GS-ADMM, each cluster updates its local estimated position value in a sequential way, which requires a global predefined order. Whereas in J-ADMM, all clusters update their local estimated position values simultaneously. Intuitively, J-ADMM is more likely to converge slower as the updating of GS-ADMM is using the most up-to-date values while J-ADMM is not. This is supported by our simulation results in Sec. VI. But for large-scale networks, updating in a sequential way may be quite time-consuming. In this way, different updating methods should be chosen according to the size of networks and other practical concerns.

## V. CONVERGENCE ANALYSIS

In this section, we analyze the convergence properties of GS-ADMM and J-ADMM. As our algorithms are applications of distributed ADMM, the analysis benefits from many existing results on general distributed ADMM [35], [43], [44].

### A. Convergence Analysis of GS-ADMM

Let  $\mathbf{p}^t = [\mathbf{p}_1^t, \mathbf{p}_2^t, \dots, \mathbf{p}_m^t]^T$  and  $\boldsymbol{\lambda}^t = [\boldsymbol{\lambda}_{i,j}^t]_{i,j,e_i,j \in E}$  be the iterates generated by algorithm GS-ADMM following (22) and (23). Assume the initial problem (12) admits a solution  $(\mathbf{p}^*, \boldsymbol{\lambda}^*)$ , then the following theorem holds:

*Theorem 1:* ([35], [43], [44]): Let  $\bar{\mathbf{p}}^{t+1} = \frac{1}{t+1} \sum_{k=0}^t \mathbf{p}^{k+1}$  be the average of  $\mathbf{p}^k$  up to iteration time  $t+1$ . If assumption 1 is satisfied, then the followings hold for all  $t$ :

$$(1) \quad 0 \leq L(\bar{\mathbf{p}}^{t+1}, \boldsymbol{\lambda}^*) - L(\mathbf{p}^*, \boldsymbol{\lambda}^*) \leq \frac{c_0}{t+1} \quad (26)$$

(2) The sequence  $(\mathbf{p}_1^t, \mathbf{p}_2^t, \dots, \mathbf{p}_m^t)$  deduced by GS-ADMM converge to  $(\mathbf{p}_1^*, \mathbf{p}_2^*, \dots, \mathbf{p}_m^*)$  and  $J_1 \mathbf{p}_1^* = J_2 \mathbf{p}_2^* = \dots = J_m \mathbf{p}_m^*$ . In other words,  $\lim_{t \rightarrow \infty} \|\mathbf{p}^t - \mathbf{p}^*\| = 0$  where  $\mathbf{p}^* = [\mathbf{p}_1^{*T}, \mathbf{p}_2^{*T}, \dots, \mathbf{p}_m^{*T}]^T$  and  $J_1 \mathbf{p}_1^* = J_2 \mathbf{p}_2^* = \dots = J_m \mathbf{p}_m^*$ .

Here  $L(\mathbf{p}, \boldsymbol{\lambda}) = f(\mathbf{p}) + \boldsymbol{\lambda}^T C J \mathbf{p}$  is the Lagrangian function (note: not the augmented Lagrangian function),

$$c_0 = \frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 + \frac{\rho}{2} (\|HJ(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \|J\mathbf{p}^0 - J\mathbf{p}^*\|^2) \quad (27)$$

and  $H = \min\{0, C\}$  ( $H_{i,j} = \min\{0, C_{i,j}\}$ ).

*Proof:* (26) can be obtained following Theorem 4.4 in [35], a detailed proof is given in the Appendix. To prove the second statement, recall that the objective function is

$$f(\mathbf{d}) = \sum_{i=1}^m \sum_{k=1}^{N_i} \sigma_{i,k}^{-2} (d_{i,k}^2 - 2d_{i,k}r_{i,k} + r_{i,k}^2)$$

Present  $h_{i,k} = \sigma_{i,k}^{-2} (d_{i,k}^2 - 2d_{i,k}r_{i,k} + r_{i,k}^2)$ , then we have  $f(\mathbf{d}) = \sum_{i=1}^m \sum_{k=1}^{N_i} h_{i,k}$ . Note that  $h_{i,k}$  is a quadratic function and is strongly convex. Since the sum of strongly convex functions is still strongly convex, our objective function  $f(\mathbf{d})$  is strongly convex. Further note that  $f(\mathbf{p})$  is equal to (change of expression

of  $f(\mathbf{d})$  and the set  $\mathcal{P}_i$  is convex and closed. Therefore, the second statement can be obtained following [43] and [44]. ■

*Remark 2:* Recall  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho C J \mathbf{p}^{k+1}$ , so we can get

$$\begin{aligned} \boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \rho C J \mathbf{p}^{k+1} \\ &= \boldsymbol{\lambda}^{k-1} + \rho C J (\mathbf{p}^{k+1} + \mathbf{p}^k) = \dots = \boldsymbol{\lambda}^0 + \rho C J \sum_{i=1}^{k+1} \mathbf{p}^i \end{aligned}$$

When  $k \rightarrow \infty$ , we have  $\boldsymbol{\lambda}^{k+1} \rightarrow \boldsymbol{\lambda}^*$ . In other words,  $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}^0 + \rho C J \sum_{i=1}^{\infty} \mathbf{p}^i$ . So  $c_0$  can be represented as:

$$c_0 = \frac{\rho}{2} \|CJ \sum_{i=1}^{\infty} \mathbf{p}^i\|^2 + \frac{\rho}{2} (\|HJ(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \|J\mathbf{p}^0 - J\mathbf{p}^*\|^2)$$

It is clear that  $c_0$  will increase with an increase in  $\rho$ , so if the iteration time  $t$  is fixed,  $L(\bar{\mathbf{p}}^{t+1}, \boldsymbol{\lambda}^*) - L(\mathbf{p}^*, \boldsymbol{\lambda}^*)$  will also increase with an increase in  $\rho$ . That is to say, with  $\rho$  increasing, the iteration time to reach convergence will be longer, namely convergence rate will be slower. This result will be confirmed by simulations in Sec VI.

Although with an increase in  $\rho$ , the convergence rate will decrease,  $\rho$  can not be too small. This is because if  $\rho$  is too small, the constraint  $J_i \mathbf{p}_i = J_j \mathbf{p}_j$  is weak, which makes reaching consistency across clusters difficult.

Directly following the statements in Theorem 1, we can obtain the following result on the convergence speed:

*Theorem 2:* The convergence rate of GS-ADMM is  $O(1/t)$ , where  $t$  is the iteration time.

*Proof:* The conclusion can be obtained directly from the proof of Theorem 1 and is omitted. ■

### B. Convergence Analysis of J-ADMM

To analyze the convergence of J-ADMM, we first define several terms: Let  $\mathbf{p}^t = [\mathbf{p}_1^t, \mathbf{p}_2^t, \dots, \mathbf{p}_m^t]^T$  and  $\boldsymbol{\lambda}^t = [\boldsymbol{\lambda}_{i,j}^t]_{i,j,e_i,j \in E}$  be the results for (24) and (25) for iteration time  $t$ . Augment the coefficients  $\gamma_i$  of proximal terms into a matrix  $Q_P = \text{diag}\{\gamma_1 I_D, \gamma_2 I_D, \dots, \gamma_m I_D\}$  and introduce a positive definite diagonal matrix  $Q_C = \text{diag}\{|B_1| I_D, |B_2| I_D, \dots, |B_m| I_D\}$ , where  $|B_i|$  is the number of clusters in  $B_i$ . Since  $Q_C$  and  $Q_P$  are both diagonal matrices, we can define a new diagonal matrix  $\bar{Q}$  according to  $\bar{Q}^T \bar{Q} = Q_C + I + Q_P$  where  $I$  is the identity matrix. It can be easily verified that  $\bar{Q}$  has the following form:

$$\bar{Q} = \text{diag}\{\gamma'_1 I_D, \gamma'_2 I_D, \dots, \gamma'_m I_D\} \quad (28)$$

with  $\gamma'_i > 0$  for  $i = 1, 2, \dots, m$ . Assuming the original problem (12) admits a solution  $(\mathbf{p}^*, \boldsymbol{\lambda}^*)$ , then we have the following theorem:

*Theorem 3:* Let  $\bar{\mathbf{p}}^{t+1} = \frac{1}{t+1} \sum_{k=0}^t \mathbf{p}^{k+1}$  be the average of  $\mathbf{p}^k$  up to iteration time  $t+1$  and denote the eigenvalues of  $C^T C$  as  $\alpha_i$ . If assumption 1 holds and  $\gamma'_i \geq \sqrt{\alpha_{\max}}$  is true with  $\alpha_{\max} = \max\{\alpha_i\}$ , then the following holds for all  $t$ :

$$0 \leq L(\bar{\mathbf{p}}^{t+1}, \boldsymbol{\lambda}^*) - L(\mathbf{p}^*, \boldsymbol{\lambda}^*) \leq \frac{c_1}{t+1} \quad (29)$$

where  $L(\mathbf{p}, \boldsymbol{\lambda}) = f(\mathbf{p}) + \boldsymbol{\lambda}^T C J \mathbf{p}$  is the Lagrangian function, and

$$c_1 = \frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 + \frac{\rho}{2} (\|\bar{Q}J(\mathbf{p}^0 - \mathbf{p}^*)\|^2) \quad (30)$$

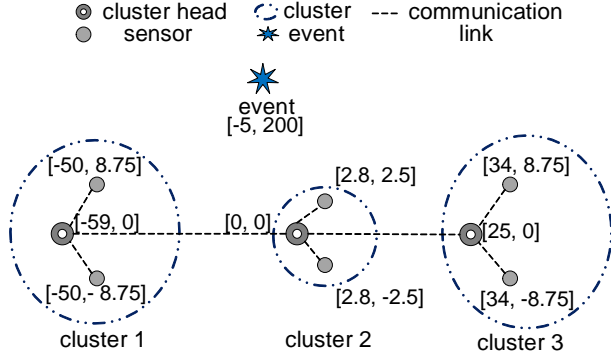


Fig. 2: Event localization structure used in simulations. The values in  $[\bullet]$  denote positions (x, y coordinates) of sensors.

*Proof:* See Appendix. ■

From Theorem 3, we can easily obtain the following results on the convergence speed:

*Theorem 4:* The convergence rate of J-ADMM is  $O(1/t)$ , where  $t$  is the iteration time.

*Proof:* The result can be obtained directly from the proof of Theorem 3 and is omitted. ■

Since  $c_0$  and  $c_1$  are of the same form, Remark 2 for GS-ADMM also apply to the J-ADMM case. Next, we use numerical experiment results to evaluate the performance of GS-ADMM and J-ADMM.

## VI. SIMULATION

We compared the localization performances of the proposed algorithms GS-ADMM and J-ADMM with the algorithm in which every cluster conducts localization independently without communications with other clusters (we call it single cluster localization and denote it as SCL). Both localization error (differences between estimated and actual event positions) and localization consistency (differences in estimated positions between clusters) are compared under different penalty parameters  $\rho$ , noise standard deviations  $\sigma_{i,k}$ , and iteration numbers  $t$ . Moreover, convergence rates of our algorithms are also evaluated. The localization setup for sensors are adopted from [45], which is a practical application of an acoustic event localization system. See Fig. 2 for the detailed spatial distribution of all sensor nodes.

To facilitate comparison of obtained simulation results, we first define two performance indexes:

*Localization Error:* we use the root mean square error (RMSE) to quantify the error between estimated and true positions for every cluster, which is denoted as  $ERR_{RMSE}$ :

$$ERR_{RMSE} = \sqrt{\frac{\sum_{j=1}^L \|\mathbf{x}_j - \mathbf{x}^*\|^2}{L}}$$

where  $L$  is the number of Monte Carlo trials,  $\mathbf{x}_j$  is the estimated position in the  $j$ th Monte Carlo trial in a certain cluster and  $\mathbf{x}^*$  is the true position of the target event.

*Localization Inconsistency:* We also use the root mean square error (RMSE) to quantify the localization inconsistency

(difference) in estimated event positions between  $m$  clusters, which is denoted as  $INC_{RMSE}$ :

$$INC_{RMSE} = \sqrt{\frac{\sum_{k=1}^L \sum_{i=1}^{m-1} \sum_{j=i+1}^m \|\mathbf{x}_{i,k} - \mathbf{x}_{j,k}\|^2}{L}}$$

where  $L$  is the number of Monte Carlo trials,  $\mathbf{x}_{i,k}$  is the estimated position obtained from the  $i$ th cluster in the  $k$ th Monte Carlo trial.  $m$  is the number of clusters.

### A. The influence of $\rho$ on localization error and localization inconsistency

In the simulation, we set the noise standard deviation  $\sigma_{i,k} = 0.05$  and varied  $\rho$  to check its influence on the localization error and localization inconsistency. The number of iteration time is fixed to 50 in all simulations and the  $ERR_{RMSE}$  and  $INC_{RMSE}$  under different  $\rho$  are summarized in Tables I and II, respectively. We denote cluster  $i$  as  $CL_i$  in the following tables:

TABLE I:  $ERR_{RMSE}$  and  $INC_{RMSE}$  of GS-ADMM vs  $\rho$

$\rho$	$ERR_{RMSE}$			$INC_{RMSE}$
	$CL_1$	$CL_2$	$CL_3$	
$10^{-5}$	1.4232	1.1452	1.3690	1.1133
$10^{-4}$	1.1046	1.0414	1.0940	0.3520
$10^{-3}$	1.0413	1.0414	1.0414	0.0033
$10^{-2}$	1.0458	1.0458	1.0458	2.68e-4
$10^{-1}$	44.5598	44.5417	44.6012	0.0199

TABLE II:  $ERR_{RMSE}$  and  $INC_{RMSE}$  of J-ADMM vs  $\rho$

$\rho$	$ERR_{RMSE}$			$INC_{RMSE}$
	$CL_1$	$CL_2$	$CL_3$	
$10^{-5}$	1.4231	1.1496	1.3691	1.1178
$10^{-4}$	1.1047	1.0414	1.0937	0.3516
$10^{-3}$	1.0414	1.0414	1.0414	9.52e-4
$10^{-2}$	1.3596	1.3563	1.3598	0.0074
$10^{-1}$	93.8179	93.7792	93.8193	0.0388

From Tables I and II, we can see that there is a certain range of parameter  $\rho$  which gives small  $ERR_{RMSE}$  and  $INC_{RMSE}$  for both GS-ADMM and J-ADMM when the iteration number is fixed to 50. When  $\rho$  is too small, e.g.,  $10^{-5}$ , the localization inconsistency of estimated event positions obtained from three clusters is larger. This is because small  $\rho$  leads to a slow rate to reach consistency across clusters. When  $\rho$  is too large, e.g.,  $10^{-1}$  used in our experiments, although its  $INC_{RMSE}$  is small, which means good consistency across clusters, its  $ERR_{RMSE}$  is unsatisfactory. This is because a large  $\rho$  leads to a slow convergence rate, which makes 50 iterations insufficient to reach convergence to the optimal value  $\mathbf{p}^*$ . In fact, as confirmed in Fig. 3 and Fig. 4, when  $\rho = 10^{-3}$ , convergence is achieved after 20 iterations for both GS-ADMM and J-ADMM, while when  $\rho = 10^{-1}$ , GS-ADMM needs about 200 iterations to achieve convergence and J-ADMM needs about 400 iterations. These results also confirmed the prediction in Remark 1, i.e., the convergence rate of GS-ADMM is faster than J-ADMM, and Remark 2, i.e., larger  $\rho$  leads to slower convergence rate.



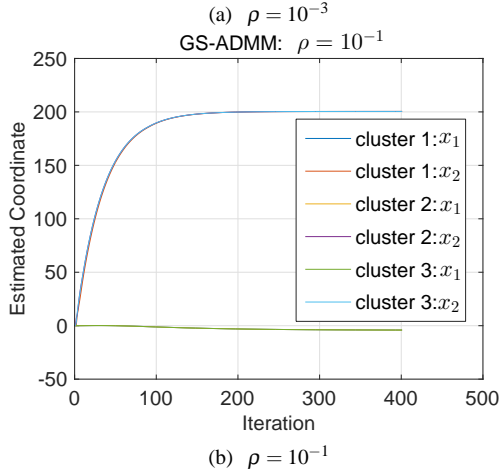
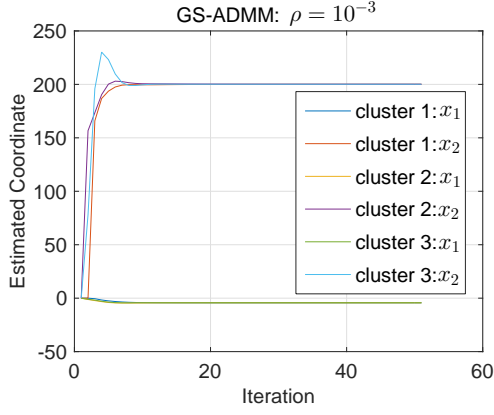


Fig. 3: The evolution of estimated positions (both x-coordinate  $x_1$  and y-coordinate  $x_2$ ) in GS-ADMM

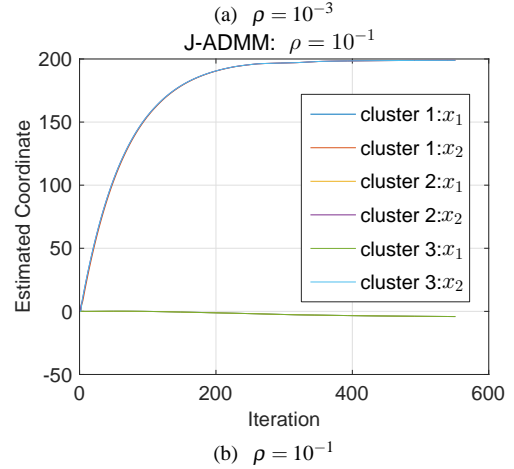
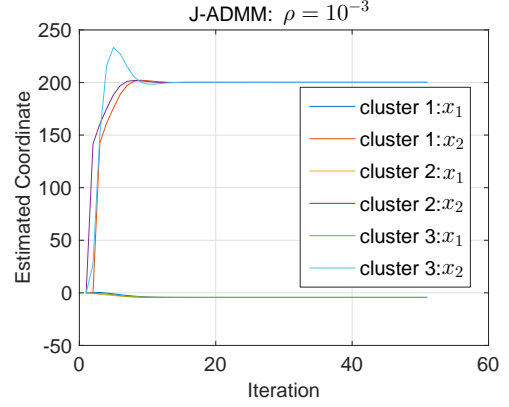


Fig. 4: The evolution of estimated positions (both x-coordinate  $x_1$  and y-coordinate  $x_2$ ) in J-ADMM

#### B. The influence of noise level on $ERR_{RMSE}$

Setting  $\rho = 10^{-3}$ , we also simulated the localization algorithms under different levels of Gaussian noise standard deviation  $\sigma_{i,k}$ . The situation in which a central node collects all clusters' information for localization is also considered, which we call total clusters localization and is named as TCL. The simulation results are summarized in Tables III, IV, and Fig. 5 to Fig. 8.

TABLE III:  $ERR_{RMSE}$  of SCL and TCL vs Noise

$\sigma_{i,k}$	SCL			TCL
	CL <sub>1</sub>	CL <sub>2</sub>	CL <sub>3</sub>	
0.00	1.29e-4	0.0338	1.34e-4	3.70e-4
0.01	0.3322	0.9242	0.3202	0.1026
0.02	0.7051	1.9314	0.6732	0.2656
0.05	1.5553	4.8132	1.4937	0.6471
0.10	3.1550	11.269	3.6144	1.1901

From Tables III, IV, we can see that both GS-ADMM and J-ADMM performed better than SCL under different noise standard deviations, especially when noise is strong. Fig. 5 to Fig. 8 visualize the estimated locations obtained from 3 clusters in 50 Monte Carlo trials for SCL, TCL, and our proposed algorithms GS-ADMM and J-ADMM under  $\rho = 10^{-3}$ . It is clear that the degree of inconsistency across clusters in SCL is larger than our algorithms. In fact, SCL has

TABLE IV:  $ERR_{RMSE}$  of GS-ADMM and J-ADMM vs Noise

$\sigma_{i,k}$	GS-ADMM			J-ADMM		
	CL <sub>1</sub>	CL <sub>2</sub>	CL <sub>3</sub>	CL <sub>1</sub>	CL <sub>2</sub>	CL <sub>3</sub>
0.00	0.0028	5.7e-4	0.0035	0.0013	0.0012	0.0011
0.01	0.2442	0.2439	0.2436	0.2440	0.2440	0.2440
0.02	0.4986	0.4987	0.4988	0.4986	0.4986	0.4986
0.05	1.0413	1.0414	1.0414	1.0414	1.0414	1.0414
0.10	2.5724	2.5723	2.5722	2.5723	2.5723	2.5724

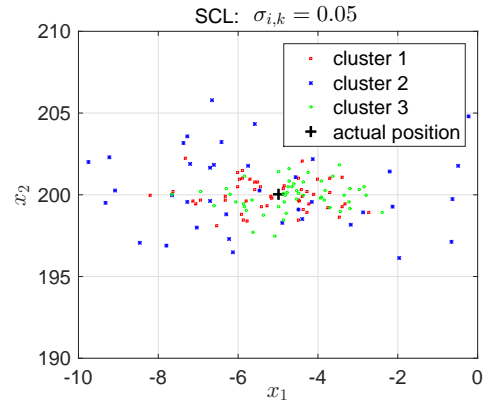


Fig. 5: SCL: Estimated event location distribution



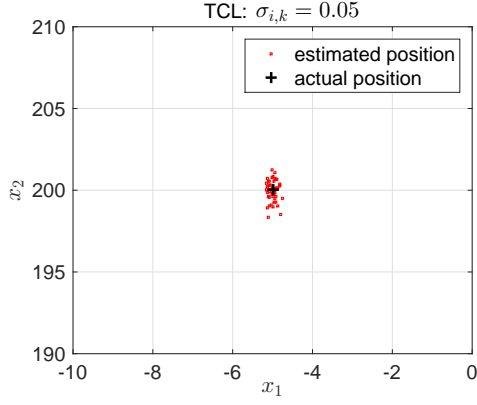


Fig. 6: TCL: Estimated event location distribution

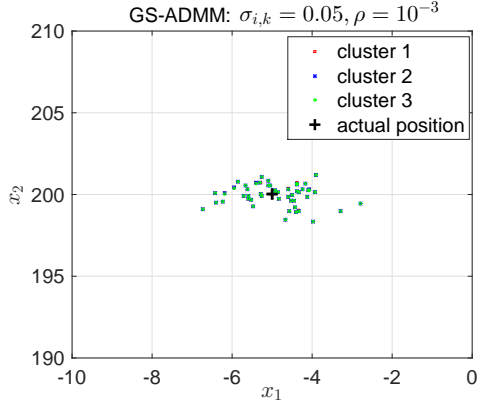


Fig. 7: GS-ADMM: Estimated event location distribution

much larger standard deviations from actual positions than our approaches (cf. Tables III and IV). In addition, the  $\text{ERR}_{\text{RMSE}}$  of cluster 2 in SCL is larger than clusters 1 and 3, which is due to the smaller distance between sensors in cluster 2.

#### C. The influence of noise level on $\text{INC}_{\text{RMSE}}$

Setting  $\rho = 10^{-3}$ , we also evaluated the influence of noise level on  $\text{INC}_{\text{RMSE}}$ . The results are summarized in Table V.

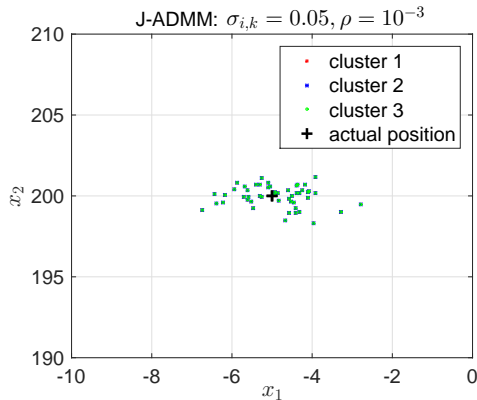


Fig. 8: J-ADMM: Estimated event location distribution

TABLE V:  $\text{INC}_{\text{RMSE}}$  of SCL, GS-ADMM, and J-ADMM vs Noise

$\sigma_{i,k}$	SCL	GS-ADMM	J-ADMM
0.00	0.0339	0.0031	1.77e-4
0.01	0.9813	0.0031	2.01e-4
0.02	2.0797	0.0032	3.74e-4
0.05	5.1190	0.0033	9.52e-4
0.10	11.421	0.0035	0.0016

Table V indicates that the proposed GS-ADMM and J-ADMM have smaller localization inconsistency ( $\text{INC}_{\text{RMSE}}$ ) compared with SCL. In addition, noise strength has smaller influence on  $\text{INC}_{\text{RMSE}}$  of our algorithms than SCL. In other words, our proposed algorithms GS-ADMM and J-ADMM can achieve good consistency across clusters even under large noise standard deviations. As indicated before, consistency is of crucial importance in many applications.

#### D. The influence of iterations on $\text{ERR}_{\text{RMSE}}$ and $\text{INC}_{\text{RMSE}}$

We also simulated the algorithms under different numbers of iterations. The results in Fig. 9, Fig. 10, and Fig. 11 show that both  $\text{ERR}_{\text{RMSE}}$  and  $\text{INC}_{\text{RMSE}}$  decrease with the iteration number in both GS-ADMM and J-ADMM, confirming the convergence properties of our proposed algorithms.

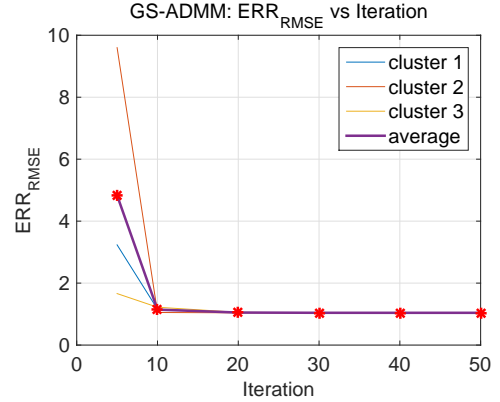


Fig. 9: GS-ADMM:  $\text{ERR}_{\text{RMSE}}$  vs Iteration

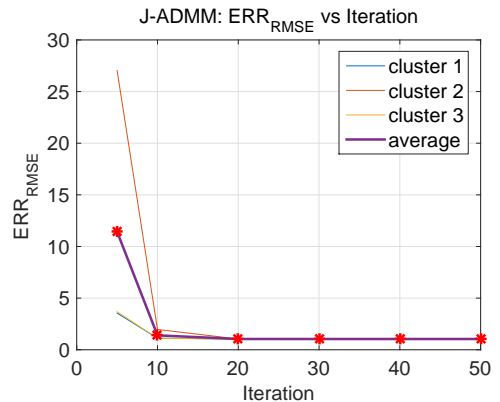


Fig. 10: J-ADMM:  $\text{ERR}_{\text{RMSE}}$  vs Iteration

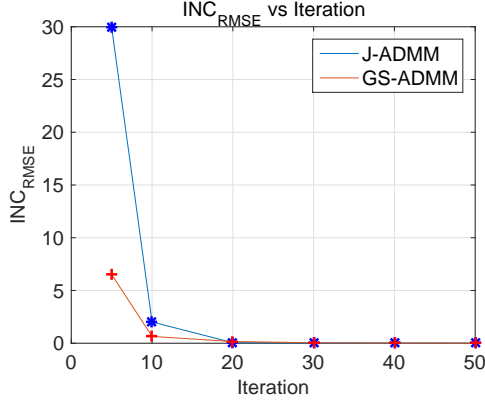


Fig. 11: INC<sub>RMSE</sub> vs Iteration

## VII. CONCLUSIONS

We proposed a cluster based decentralized event localization structure that can distribute the complicated localization optimization to multiple sensor nodes. The structure can guarantee consistency among all sensor nodes with moderate communication overhead. Then we proposed two ADMM based distributed algorithms, i.e., GS-ADMM and J-ADMM, for distributed event localization under the proposed structure. Convergence properties of the algorithms are analyzed theoretically. Numerical simulations showed that the proposed algorithms are robust to measurement noises. Furthermore, numerical simulations also confirmed that the proposed algorithms are superior to the localization approach with isolated clusters in terms of both localization accuracy and localization consistency.

## APPENDIX

### A. Proof of (26) in Theorem 1

To prove (26) in Theorem 1, we first introduce two lemmas:

*Lemma 1:* Let  $\mathbf{p}^t = [\mathbf{p}_1^t, \mathbf{p}_2^t, \dots, \mathbf{p}_m^t]^T$  and  $\boldsymbol{\lambda}^t = [\boldsymbol{\lambda}_{i,j}^t]_{ij, e_{i,j} \in E}$  be the iterates generated by GS-ADMM following (22) and (23), then the following inequality holds for all  $k$ :

$$f(\mathbf{p}) - f(\mathbf{p}^{k+1}) + (\mathbf{p} - \mathbf{p}^{k+1})^T J^T C^T \boldsymbol{\lambda}^{k+1} + \rho(\mathbf{p} - \mathbf{p}^{k+1})^T J^T (-C^T H + H^T H + I) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \geq 0 \quad (31)$$

where  $C$  is the edge-node incident matrix defined in (15),  $H = \min\{0, C\}$  and  $I$  is the identity matrix.

*Proof:* Denote by  $g_i$  the function

$$g_i^k(\mathbf{p}_i) = \sum_{j \in \hat{B}_i, j \geq i} (\boldsymbol{\lambda}_{i,j}^{kT} (J_i \mathbf{p}_i - J_j \mathbf{p}_j^k) + \frac{\rho}{2} \|J_i \mathbf{p}_i - J_j \mathbf{p}_j^k\|^2) \quad (32) \\ + \sum_{j \in \hat{B}_i, j < i} (\boldsymbol{\lambda}_{i,j}^{kT} (J_i \mathbf{p}_i - J_j \mathbf{p}_j^{k+1}) + \frac{\rho}{2} \|J_i \mathbf{p}_i - J_j \mathbf{p}_j^{k+1}\|^2)$$

From the update of (22), we know  $\mathbf{p}_i^{k+1}$  is the optimizer of  $g_i^k + f_i$ . Suppose  $h(\mathbf{p}_i^{k+1}) \in \partial f_i(\mathbf{p}_i^{k+1})$ , then we can get  $h(\mathbf{p}_i^{k+1}) + \nabla g_i(\mathbf{p}_i^{k+1}) = 0$ , and thus  $(\mathbf{p}_i - \mathbf{p}_i^{k+1})^T [h(\mathbf{p}_i^{k+1}) + \nabla g_i(\mathbf{p}_i^{k+1})] = 0$ . On the other hand, as  $f_i$  is convex, we have the following relationship:

$$f_i(\mathbf{p}_i) \geq f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T h(\mathbf{p}_i^{k+1})$$

Then we can get

$$f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \nabla g_i(\mathbf{p}_i^{k+1}) \geq 0$$

Substituting  $\nabla g_i(\mathbf{p}_i^{k+1})$  with (32), we have

$$f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \cdot \\ \left( \sum_{j \in \hat{B}_i, j \geq i} (J_i^T \boldsymbol{\lambda}_{i,j}^k + \rho J_i^T (J_j \mathbf{p}_i^{k+1} - J_j \mathbf{p}_j^k)) \right) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \cdot \\ \left( \sum_{j \in \hat{B}_i, j < i} (J_i^T \boldsymbol{\lambda}_{i,j}^k + \rho J_i^T (J_j \mathbf{p}_i^{k+1} - J_j \mathbf{p}_j^{k+1})) \right) \geq 0$$

Noting  $\boldsymbol{\lambda}_{i,i} = 0$ , using (23) leads to

$$f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \cdot \\ \left( \sum_{j \in \hat{B}_i} J_i^T \boldsymbol{\lambda}_{i,j}^{k+1} + \sum_{j \in \hat{B}_i, j \geq i} \rho J_i^T (J_j \mathbf{p}_j^{k+1} - J_j \mathbf{p}_j^k) \right) \geq 0$$

Noting  $\boldsymbol{\lambda}_{i,j} = -\boldsymbol{\lambda}_{j,i}$ , from the definition of  $C$ , we can rewrite the above inequality as

$$f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \cdot \\ (J_i^T [C]_i^T \boldsymbol{\lambda}^{k+1} + \sum_{j \in \hat{B}_i, j \geq i} \rho J_i^T (J_j \mathbf{p}_j^{k+1} - J_j \mathbf{p}_j^k)) \geq 0 \quad (33)$$

here  $[C]_i$  denotes the column of  $C$  associated with cluster  $i$ .

By summing the above relation over  $i = 1, 2, \dots, m$ , the following two equations can be obtained:

$$\sum_{i=1}^m (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T J_i^T [C]_i^T \boldsymbol{\lambda}^{k+1} = (J\mathbf{p} - J\mathbf{p}^{k+1})^T C^T \boldsymbol{\lambda}^{k+1} \\ \sum_{i=1}^m (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \left( \sum_{j \in \hat{B}_i, j \geq i} \rho J_i^T (J_j \mathbf{p}_j^{k+1} - J_j \mathbf{p}_j^k) \right) \\ = \rho (J\mathbf{p} - J\mathbf{p}^{k+1})^T [(-C + H)^T H + I] (J\mathbf{p}^{k+1} - J\mathbf{p}^k)$$

Combining the above two equalities with (33), we can get the lemma. ■

*Lemma 2:* Let  $\mathbf{p}^t = [\mathbf{p}_1^t, \mathbf{p}_2^t, \dots, \mathbf{p}_m^t]^T$  and  $\boldsymbol{\lambda}^t = [\boldsymbol{\lambda}_{i,j}^t]_{ij, e_{i,j} \in E}$  be the iterates generated by GS-ADMM following (22) and (23), then the following inequality holds for all  $k$ :

$$-(J\mathbf{p}^{k+1})^T C^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) \quad (34) \\ + \rho (J\mathbf{p}^* - J\mathbf{p}^{k+1})^T (H^T H - C^T H + I) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \\ = -\frac{1}{2\rho} (\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 - \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2) \\ -\frac{\rho}{2} (\|HJ(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 - \|HJ(\mathbf{p}^k - \mathbf{p}^*)\|^2) \\ -\frac{\rho}{2} (\|J\mathbf{p}^{k+1} - J\mathbf{p}^*\|^2 - \|J\mathbf{p}^k - J\mathbf{p}^*\|^2) \\ -\frac{\rho}{2} \|HJ(\mathbf{p}^{k+1} - \mathbf{p}^k) - CJ\mathbf{p}^{k+1}\|^2 - \frac{\rho}{2} \|J\mathbf{p}^{k+1} - J\mathbf{p}^k\|^2$$

*Proof:* Since for a scalar  $a$ ,  $a^T = a$  holds, and recall  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho CJ\mathbf{p}^{k+1}$ , we can get

$$(\mathbf{p}^{k+1})^T J^T C^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) = \frac{1}{\rho} (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k)^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) \quad (35)$$

In addition, as  $(\mathbf{p}^*, \boldsymbol{\lambda}^*)$  is the saddle point, we have  $CJ\mathbf{p}^* = 0$ . So we can establish the following relationships using algebra manipulation:

$$(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k)^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) = \frac{1}{2} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 \quad (36)$$

$$+ \frac{1}{2} (\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 - \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2)$$

$$(\mathbf{p}^{k+1} - \mathbf{p}^*)^T J^T I J (\mathbf{p}^{k+1} - \mathbf{p}^k) = \frac{1}{2} \|J\mathbf{p}^{k+1} - J\mathbf{p}^k\|^2 \quad (37)$$

$$+ \frac{1}{2} (\|J\mathbf{p}^{k+1} - J\mathbf{p}^*\|^2 - \|J\mathbf{p}^k - J\mathbf{p}^*\|^2)$$

$$(\mathbf{p}^{k+1} - \mathbf{p}^*)^T J^T H^T H J (\mathbf{p}^{k+1} - \mathbf{p}^k) \quad (38)$$

$$= \frac{1}{2} (\|HJ(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 - \|HJ(\mathbf{p}^k - \mathbf{p}^*)\|^2)$$

$$+ \frac{1}{2} \|HJ(\mathbf{p}^{k+1} - \mathbf{p}^k)\|^2$$

$$(\mathbf{p}^{k+1} - \mathbf{p}^*)^T J^T C^T H J (\mathbf{p}^{k+1} - \mathbf{p}^k) \quad (39)$$

$$= \frac{1}{2} \|HJ(\mathbf{p}^{k+1} - \mathbf{p}^k)\|^2 + \frac{1}{2\rho^2} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2$$

$$- \frac{1}{2} \|HJ(\mathbf{p}^{k+1} - \mathbf{p}^k) - CJ\mathbf{p}^{k+1}\|^2$$

Then (34) can be proven by plugging equations (35) to (39) into the left part of (34). ■

Now we proceed to prove Theorem 1. Set  $\mathbf{p} = \mathbf{p}^*$  in (31), and recall  $CJ\mathbf{p}^* = 0$ , then we have

$$f(\mathbf{p}^*) - f(\mathbf{p}^{k+1}) - \mathbf{p}^{(k+1)T} J^T C^T \boldsymbol{\lambda}^{k+1} \quad (40)$$

$$+ \rho(\mathbf{p}^* - \mathbf{p}^{k+1})^T J^T (-C^T H + H^T H + I) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \geq 0$$

Adding and subtracting the term  $\boldsymbol{\lambda}^{*T} CJ\mathbf{p}^{k+1}$  from the left side of (40), we can get

$$f(\mathbf{p}^*) - f(\mathbf{p}^{k+1}) - \boldsymbol{\lambda}^{*T} CJ\mathbf{p}^{k+1} - \mathbf{p}^{(k+1)T} J^T C^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*)$$

$$+ \rho(\mathbf{p}^* - \mathbf{p}^{k+1})^T J^T (-C^T H + H^T H + I) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \geq 0$$

Now by applying (34) into the above inequality, the following inequality can be obtained:

$$f(\mathbf{p}^*) - f(\mathbf{p}^{k+1}) - \boldsymbol{\lambda}^{*T} CJ\mathbf{p}^{k+1}$$

$$- \frac{1}{2\rho} (\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 - \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2)$$

$$- \frac{\rho}{2} (\|HJ(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 - \|HJ(\mathbf{p}^k - \mathbf{p}^*)\|^2)$$

$$- \frac{\rho}{2} \|HJ(\mathbf{p}^{k+1} - \mathbf{p}^k) - CJ\mathbf{p}^{k+1}\|^2 - \frac{\rho}{2} \|J\mathbf{p}^{k+1} - J\mathbf{p}^k\|^2$$

$$- \frac{\rho}{2} (\|J\mathbf{p}^{k+1} - J\mathbf{p}^*\|^2 - \|J\mathbf{p}^k - J\mathbf{p}^*\|^2) \geq 0$$

Summing the inequality from  $k = 1$  to  $t$ , we can obtain the following inequality:

$$(t+1)f(\mathbf{p}^*) - \sum_{k=0}^t f(\mathbf{p}^{k+1}) - \boldsymbol{\lambda}^{*T} CJ \sum_{k=0}^t \mathbf{p}^{k+1}$$

$$+ \frac{\rho}{2} (\|HJ(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \|J\mathbf{p}^0 - J\mathbf{p}^*\|^2) + \frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2$$

$$\geq \sum_{k=0}^t \frac{\rho}{2} (\|HJ(\mathbf{p}^{k+1} - \mathbf{p}^k) - CJ\mathbf{p}^{k+1}\|^2 + \|J\mathbf{p}^{k+1} - J\mathbf{p}^k\|^2)$$

$$+ \frac{\rho}{2} (\|HJ(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 + \|J\mathbf{p}^{k+1} - J\mathbf{p}^*\|^2)$$

$$+ \frac{1}{2\rho} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 \geq 0$$

In addition, as our function is strongly convex, we have  $\sum_{k=0}^t f(\mathbf{p}^{k+1}) \geq (t+1)f(\bar{\mathbf{p}}^{t+1})$ , then we can get

$$(t+1)f(\mathbf{p}^*) - (t+1)f(\bar{\mathbf{p}}^{t+1}) - (t+1)\boldsymbol{\lambda}^{*T} CJ\bar{\mathbf{p}}^{t+1} +$$

$$\frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 + \frac{\rho}{2} (\|HJ(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \|J\mathbf{p}^0 - J\mathbf{p}^*\|^2) \geq 0$$

Dividing  $-(t+1)$  on both sides yields

$$f(\bar{\mathbf{p}}^{t+1}) + \boldsymbol{\lambda}^{*T} CJ\bar{\mathbf{p}}^{t+1} - f(\mathbf{p}^*) \leq \frac{1}{t+1} \cdot \quad (41)$$

$$(\frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 + \frac{\rho}{2} (\|HJ(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \|J\mathbf{p}^0 - J\mathbf{p}^*\|^2))$$

Combining the above relationship (41) with the definition of Lagrangian function, Theorem 1 is proven. ■

### B. Proof of Theorem 3

To prove Theorem 3, we first introduce two lemmas:

*Lemma 3:* Let  $\mathbf{p}^t = [\mathbf{p}_1^t, \mathbf{p}_2^t, \dots, \mathbf{p}_m^t]^T$  and  $\boldsymbol{\lambda}^t = [\boldsymbol{\lambda}_{i,j}^t]_{i,j \in E}$  be the iterates generated by J-ADMM following (24) and (25), then the following inequality holds for all  $k$ :

$$f(\mathbf{p}) - f(\mathbf{p}^{k+1}) + (\mathbf{p} - \mathbf{p}^{k+1})^T J^T C^T \boldsymbol{\lambda}^{k+1} \quad (42)$$

$$+ \rho(\mathbf{p} - \mathbf{p}^{k+1})^T J^T (-C^T C + \bar{Q}^T \bar{Q}) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \geq 0$$

where  $\bar{Q}$  is defined in (28).

*Proof:* Denote by  $g_i$  the function

$$g_i^k(\mathbf{p}_i) = \sum_{j \in \tilde{B}_i} (\boldsymbol{\lambda}_{i,j}^{kT} (J_i \mathbf{p}_i - J_j \mathbf{p}_j^k)) \quad (43)$$

$$+ \frac{\rho}{2} \|J_i \mathbf{p}_i - J_j \mathbf{p}_j^k\|^2 + \frac{\rho \gamma_i}{2} \|J_i \mathbf{p}_i - J_i \mathbf{p}_i^k\|^2$$

From the update of (24), we know  $\mathbf{p}_i^{k+1}$  is the optimizer of  $g_i^k + f_i$ . Suppose  $h(\mathbf{p}_i^{k+1}) \in \partial f_i(\mathbf{p}_i^{k+1})$ , then we can get  $h(\mathbf{p}_i^{k+1}) + \nabla g_i(\mathbf{p}_i^{k+1}) = 0$ , and thus  $(\mathbf{p}_i - \mathbf{p}_i^{k+1})^T [h(\mathbf{p}_i^{k+1}) + \nabla g_i(\mathbf{p}_i^{k+1})] = 0$ . On the other hand, as  $f_i$  is convex, the following relationship holds:

$$f_i(\mathbf{p}_i) \geq f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T h(\mathbf{p}_i^{k+1})$$

Then we can get

$$f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \nabla g_i(\mathbf{p}_i^{k+1}) \geq 0$$

Substituting  $\nabla g_i(\mathbf{p}_i^{k+1})$  with (43), we can get

$$f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T J_i^T \cdot$$

$$(\sum_{j \in \tilde{B}_i} (\boldsymbol{\lambda}_{i,j}^k + \rho(J_i \mathbf{p}_i^{k+1} - J_j \mathbf{p}_j^k)) + \rho \gamma_i J_i (\mathbf{p}_i^{k+1} - \mathbf{p}_i^k)) \geq 0$$

Noting  $\boldsymbol{\lambda}_{i,i} = 0$  and using (25) we can rewrite the above inequality as

$$f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T J_i^T \cdot$$

$$(\sum_{j \in \tilde{B}_i} \boldsymbol{\lambda}_{i,j}^{k+1} + \sum_{j \in \tilde{B}_i} \rho J_j (\mathbf{p}_j^{k+1} - \mathbf{p}_j^k) + \rho \gamma_i J_i (\mathbf{p}_i^{k+1} - \mathbf{p}_i^k)) \geq 0$$

Based on the definition of matrix  $C$  and the relationship  $\boldsymbol{\lambda}_{i,j} = -\boldsymbol{\lambda}_{j,i}$ , the above inequality can be written as

$$f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T J_i^T \cdot \quad (44)$$

$$([C]_i^T \boldsymbol{\lambda}^{k+1} + \sum_{j \in \tilde{B}_i} \rho J_j (\mathbf{p}_j^{k+1} - \mathbf{p}_j^k) + \rho \gamma_i J_i (\mathbf{p}_i^{k+1} - \mathbf{p}_i^k)) \geq 0$$

Summing the above relation over  $i = 1, 2, \dots, m$  gives the following two relations:

$$\begin{aligned} & \sum_{i=1}^m (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T J_i^T \rho \left( \sum_{j \in \tilde{B}_i} J_j (\mathbf{p}_j^{k+1} - \mathbf{p}_j^k) + \gamma_i J_i (\mathbf{p}_i^{k+1} - \mathbf{p}_i^{k+1}) \right) \\ &= \rho (\mathbf{p} - \mathbf{p}^{k+1})^T J^T [-C^T C + Q_C + I + Q_P] J (\mathbf{p}^{k+1} - \mathbf{p}^k) \\ & \sum_{i=1}^m (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T J_i^T [C]_i^T \boldsymbol{\lambda}^{k+1} = (\mathbf{p} - \mathbf{p}^{k+1})^T J^T C^T \boldsymbol{\lambda}^{k+1} \end{aligned}$$

Combining the above relations with (44) gives the lemma. ■

*Lemma 4:* Let  $\mathbf{p}^t = [\mathbf{p}_1^t, \mathbf{p}_2^t, \dots, \mathbf{p}_m^t]^T$  and  $\boldsymbol{\lambda}^t = [\lambda_{i,j}^t]_{i,j \in E}$  be the iterates generated by J-ADMM following (24) and (25). Then the following inequality holds for all  $k$ :

$$\begin{aligned} & -(\mathbf{p}^{k+1})^T J^T C^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) \quad (45) \\ & + \rho (\mathbf{p}^* - \mathbf{p}^{k+1})^T J^T (-C^T C + \bar{Q}^T \bar{Q}) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \\ &= -\frac{1}{2\rho} (\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 - \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2) \\ & + \frac{\rho}{2} (\|CJ(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 - \|CJ(\mathbf{p}^k - \mathbf{p}^*)\|^2) \\ & - \frac{\rho}{2} (\|\bar{Q}J(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 - \|\bar{Q}J(\mathbf{p}^k - \mathbf{p}^*)\|^2) \\ & + \frac{\rho}{2} \|CJ(\mathbf{p}^{k+1} - \mathbf{p}^k)\|^2 - \frac{\rho}{2} \|\bar{Q}J(\mathbf{p}^{k+1} - \mathbf{p}^k)\|^2 \\ & - \frac{1}{2\rho} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 \end{aligned}$$

*Proof:* The proof is similar to the proof of Lemma 2 and is omitted. ■

Now we proceed to prove Theorem 3.

Setting  $\mathbf{p} = \mathbf{p}^*$  in (42), we can get

$$\begin{aligned} & f(\mathbf{p}^*) - f(\mathbf{p}^{k+1}) + (\mathbf{p}^* - \mathbf{p}^{k+1})^T J^T C^T \boldsymbol{\lambda}^{k+1} \\ & + \rho (\mathbf{p}^* - \mathbf{p}^{k+1})^T J^T (-C^T C + \bar{Q}^T \bar{Q}) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \geq 0 \end{aligned}$$

As  $(\mathbf{p}^*, \boldsymbol{\lambda}^*)$  is the saddle point, we have  $CJ\mathbf{p}^* = 0$ , then the above equation can be rewritten as

$$\begin{aligned} & f(\mathbf{p}^*) - f(\mathbf{p}^{k+1}) - \mathbf{p}^{(k+1)T} J^T C^T \boldsymbol{\lambda}^{k+1} \quad (46) \\ & + \rho (\mathbf{p}^* - \mathbf{p}^{k+1})^T J^T (-C^T C + \bar{Q}^T \bar{Q}) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \geq 0 \end{aligned}$$

Adding and subtracting the term  $\boldsymbol{\lambda}^{*T} CJ\mathbf{p}^{k+1}$  from the left side of (46), we can get

$$\begin{aligned} & f(\mathbf{p}^*) - f(\mathbf{p}^{k+1}) - \boldsymbol{\lambda}^{*T} CJ\mathbf{p}^{k+1} - \mathbf{p}^{(k+1)T} J^T C^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) \\ & + \rho (\mathbf{p}^* - \mathbf{p}^{k+1})^T J^T (-C^T C + \bar{Q}^T \bar{Q}) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \geq 0 \end{aligned}$$

Now by applying (45) into the above inequality, we can obtain the following inequality:

$$\begin{aligned} & f(\mathbf{p}^*) - f(\mathbf{p}^{k+1}) - \boldsymbol{\lambda}^{*T} CJ\mathbf{p}^{k+1} \\ & - \frac{1}{2\rho} (\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 - \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2) \\ & + \frac{\rho}{2} (\|CJ(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 - \|CJ(\mathbf{p}^k - \mathbf{p}^*)\|^2) \\ & - \frac{\rho}{2} (\|\bar{Q}J(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 - \|\bar{Q}J(\mathbf{p}^k - \mathbf{p}^*)\|^2) \\ & + \frac{\rho}{2} \|CJ(\mathbf{p}^{k+1} - \mathbf{p}^k)\|^2 - \frac{\rho}{2} \|\bar{Q}J(\mathbf{p}^{k+1} - \mathbf{p}^k)\|^2 \\ & - \frac{1}{2\rho} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 \geq 0 \end{aligned}$$

Summing the inequality from  $k = 1$  to  $t$ , the following inequality can be obtained:

$$\begin{aligned} & (t+1)f(\mathbf{p}^*) - \sum_{k=0}^t f(\mathbf{p}^{k+1}) - \boldsymbol{\lambda}^{*T} CJ \sum_{k=0}^t \mathbf{p}^{k+1} \\ & + \frac{\rho}{2} \|\bar{Q}J(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 \\ & \geq \frac{\rho}{2} \|CJ(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \frac{1}{2\rho} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 \\ & + \sum_{k=0}^t \frac{\rho}{2} (\|\bar{Q}J(\mathbf{p}^{k+1} - \mathbf{p}^k)\|^2 - \|CJ(\mathbf{p}^{k+1} - \mathbf{p}^k)\|^2) \\ & + \frac{\rho}{2} (\|\bar{Q}J(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 - \|CJ(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2) \\ & + \sum_{k=0}^t \frac{1}{2\rho} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 \\ & \geq \sum_{k=0}^t \frac{\rho}{2} (\|\bar{Q}J(\mathbf{p}^{k+1} - \mathbf{p}^k)\|^2 - \|C\|^2 \|J\mathbf{p}^{k+1} - J\mathbf{p}^k\|^2) \\ & + \frac{\rho}{2} (\|\bar{Q}J(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 - \|C\|^2 \|J\mathbf{p}^{k+1} - J\mathbf{p}^*\|^2) \end{aligned}$$

Since  $\|C\|^2 = \alpha_{\max}$ ,  $\bar{Q}$  is a diagonal matrix with  $\gamma'_i \geq \sqrt{\alpha_{\max}}$ , we can get that the right side of the above inequality is larger than 0, which leads to

$$\begin{aligned} & (t+1)f(\mathbf{p}^*) - \sum_{k=0}^t f(\mathbf{p}^{k+1}) - \boldsymbol{\lambda}^{*T} CJ \sum_{k=0}^t \mathbf{p}^{k+1} \\ & + \frac{\rho}{2} \|\bar{Q}J(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 \geq 0 \end{aligned}$$

In addition, as our function is strongly convex, we have  $\sum_{k=0}^t f(\mathbf{p}^{k+1}) \geq (t+1)f(\bar{\mathbf{p}}^{t+1})$  and

$$\begin{aligned} & (t+1)f(\mathbf{p}^*) - (t+1)f(\bar{\mathbf{p}}^{t+1}) - (t+1)\boldsymbol{\lambda}^{*T} CJ\bar{\mathbf{p}}^{t+1} \\ & + \frac{\rho}{2} \|\bar{Q}J(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 \geq 0 \end{aligned}$$

By dividing  $-(t+1)$  on both sides, we can obtain

$$\begin{aligned} & f(\bar{\mathbf{p}}^{t+1}) + \boldsymbol{\lambda}^{*T} CJ\bar{\mathbf{p}}^{t+1} - f(\mathbf{p}^*) \quad (47) \\ & \leq \frac{1}{t+1} \left( \frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 + \frac{\rho}{2} \|\bar{Q}J(\mathbf{p}^0 - \mathbf{p}^*)\|^2 \right) \end{aligned}$$

Combining the above relationship (47) with the definition of Lagrangian function, we can get Theorem 3. ■

## ACKNOWLEDGMENT

We would like to thanks Dr. Andrea Simonetto for providing Matlab codes for his paper [25].

## REFERENCES

- [1] P. Bonnet, J. Gehrke, and P. Seshadri. Querying the physical world. *IEEE Personal Communications*, 7(5):10–15, 2000.
- [2] V. Cevher, M. F. Duarte, and R. G. Baraniuk. Distributed target localization via spatial sparsity. In *Proceedings of the 16th European Signal Processing Conference*, pages 1–5, 2008.
- [3] J. Heidemann, Y. Li, A. Syed, J. Wills, and W. Ye. Underwater sensor networking: Research challenges and potential applications. *Technical Report ISI-TR-2005-603, USC/Information Sciences Institute*, 2005.

- [4] J. George and L. M. Kaplan. Shooter localization using a wireless sensor network of soldier-worn gunfire detection systems. *Journal of Advances in Information Fusion*, 8(1):15–32, 2013.
- [5] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusz, A. Nádas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 1–12, 2004.
- [6] S. Deligeorgis, G. Cakiades, J. George, Y. Q. Wang, and F. J. Doyle. A mobile self synchronizing smart sensor array for detection and localization of impulsive threat sources. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 351–356, 2015.
- [7] S. Nabavi, J. H. Zhang, and A. Chakraborty. Distributed optimization algorithms for wide-area oscillation monitoring in power systems using interregional PMU-PDC architectures. *IEEE Transactions on Smart Grid*, 6(5):2529–2538, 2015.
- [8] R. Hasan, R. Bobba, and H. Khurana. Analyzing NASPInet data flows. In *Proceedings of IEEE Power Systems Conference and Exposition*, pages 1–6, 2009.
- [9] L. M. Kaplan, Q. Le, and P. Molnár. Maximum likelihood methods for bearings-only target localization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3001–3004, 2001.
- [10] R. O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280, 1986.
- [11] M. Gavish and A. J. Weiss. Performance analysis of bearing-only target location algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 28(3):817–828, 1992.
- [12] G. Zhu and J. Hu. A distributed continuous-time algorithm for network localization using angle-of-arrival information. *Automatica*, 50(1):53–63, 2014.
- [13] Y. T. Chan, H. Y. C. Hang, and P. C. Ching. Exact and approximate maximum likelihood localization algorithms. *IEEE Transactions on Vehicular Technology*, 55(1):10–16, 2006.
- [14] K. Yang, G. Wang, and Z. Q. Luo. Efficient convex relaxation methods for robust target localization by a sensor network using time differences of arrivals. *IEEE Transactions on Signal Processing*, 57(7):2775–2784, 2009.
- [15] K. C. Ho and W. Xu. An accurate algebraic solution for moving source location using TDOA and FDOA measurements. *IEEE Transactions on Signal Processing*, 52(9):2453–2463, 2004.
- [16] L. Doherty, K. Pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *Proceedings of the Twentieth IEEE Annual Joint Conference of the IEEE Computer and Communications*, volume 3, pages 1655–1663, 2001.
- [17] Y. Shang and W. Rum. Improved MDS-based localization. In *Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications*, volume 4, pages 2640–2651, 2004.
- [18] J. Bachrach and C. Taylor. Localization in sensor networks. *Handbook of sensor networks: Algorithms and Architectures*, 1, 2005.
- [19] J. D. McLurkin. *Algorithms for distributed sensor networks*. PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 1999.
- [20] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Anchor-free distributed localization in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 340–341, 2003.
- [21] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AOA. In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, volume 3, pages 1734–1743, 2003.
- [22] L. Meertens and S. Fitzpatrick. The distributed construction of a global coordinate system in a network of static computational nodes from inter-node distances. *Kestrel Institute TR KES. U*, 4, 2004.
- [23] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61. ACM, 2004.
- [24] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- [25] A. Simonetto and G. Leus. Distributed maximum likelihood sensor network localization. *IEEE Transactions on Signal Processing*, 62(6):1424–1437, 2014.
- [26] G. Destino and G. Abreu. On the maximum likelihood approach for source and network localization. *IEEE Transactions on Signal Processing*, 59(10):4954–4970, 2011.
- [27] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54, 2004.
- [28] P. Biswas, T. C. Lian, T. C. Wang, and Y. Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks*, 2(2):188–220, 2006.
- [29] K. W. K. Lui, W. K. Ma, H. C. So, and F. K. W. Chan. Semidefinite programming algorithms for sensor network node localization with uncertainties in anchor positions and/or propagation speed. *IEEE Transactions on Signal Processing*, 57(2):752–763, 2009.
- [30] P. Tseng. Second-order cone programming relaxation of sensor network localization. *SIAM Journal on Optimization*, 18(1):156–185, 2007.
- [31] J. Nie. Sum of squares method for sensor network localization. *Computational Optimization and Applications*, 43(2):151–179, 2009.
- [32] P. Oguz-Ekim, J. Gomes, J. Xavier, and P. Oliveira. A convex relaxation for approximate maximum-likelihood 2d source localization from range measurements. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, pages 2698–2701, 2010.
- [33] D. V. Ouellette. Schur complements and statistics. *Linear Algebra and its Applications*, 36:187–295, 1981.
- [34] J. A. Bondy and U. S. R. Murty. *Graph theory with applications*, volume 290. Macmillan London, 1976.
- [35] E. Wei and A. Ozdaglar. Distributed alternating direction method of multipliers. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 5445–5450, 2012.
- [36] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. Johns Hopkins University Press, 2012.
- [37] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [38] B. S. He, M. Tao, and X. M. Yuan. Alternating direction method with gaussian back substitution for separable convex programming. *SIAM Journal on Optimization*, 22(2):313–340, 2012.
- [39] M. Hong and Z. Q. Luo. On the linear convergence of the alternating direction method of multipliers. *arXiv preprint arXiv:1208.3922*, 2012.
- [40] M. Tao and X. M. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, 21(1):57–81, 2011.
- [41] J. Löfberg. Yalmip: A toolbox for modeling and optimization in matlab. In *Proceedings of IEEE International Symposium on Computer Aided Control Systems Design*, pages 284–289, 2004.
- [42] W. Deng, M. Lai, Z. Peng, and W. Yin. Parallel multi-block ADMM with  $\mathcal{O}(1/k)$  convergence. *arXiv preprint arXiv:1312.3040*, 2013.
- [43] D. Han and X. Yuan. A note on the alternating direction method of multipliers. *Journal of Optimization Theory and Applications*, 155(1):227–238, 2012.
- [44] J. Mota, J. Xavier, P. Aguiar, and M. Puschel. D-ADMM: A communication-efficient distributed algorithm for separable optimization. *IEEE Transactions on Signal Processing*, 61(10):2718–2723, 2013.
- [45] F. Núñez, Y. Q. Wang, D. Grasing, S. Desai, G. Cakiades, and F. J. Doyle. Pulse-coupled time synchronization for distributed acoustic event detection using wireless sensor networks. *Working paper*, 2016.